UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,597	12/09/2003	Robert Little	60001.0270US01/MS303919.1	4741

7590 02/24/2006

Christopher J. Leonard
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
RUTLEDGE, AMELIA L	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 02/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/731,597	LITTLE ET AL.	
	Examiner	Art Unit	
	Amelia Rutledge	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *09 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: original application, filed 12/09/2003.

2. Claims 1-18 are pending in the case. Claims 1, 3, 5, and 7 are independent claims.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 1-18 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

In regard to independent claims 1, 3, and 5, claims 1, 3, and 5 cite *A programmable object model for accessing the resources of an Extensible Markup Language (XML) schema library, comprising: an application programming interface for allowing a user to programmatically access resources identified in an XML schema library*. As claimed, the *programmable object model* and *application programming interface* (API) as well as the rest of the claim limitations, are nonfunctional descriptive material for at least two reasons. First, nonfunctional descriptive material is claimed because the invention is not recorded on a computer readable medium. Secondly, even if the invention were recorded on a computer readable medium, it would not be statutory because no requisite functionality is present to satisfy the practical application

requirement. See *Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility*, p. 1-2.

Specifically, a *programmable object model* was known in the art to be: 1. the structural foundation for an object-oriented language... This foundation includes such principles as abstraction, concurrency, encapsulation, hierarchy, persistence, polymorphism, and typing... 2. The structural foundation for an object-oriented design. 3. The structural foundation for an object-oriented application (Microsoft Computer Dictionary, copyright 2002, Microsoft Corporation, p. 372). And, an *API* was known in the art as: A set of routines used by an application program to direct the performance of procedures by the computer's operating system (Microsoft Computer Dictionary, copyright 2002, Microsoft Corporation, p. 33). At best, the claimed invention describes a model or abstract foundation of ideas and a set of program routines which represents a manipulation of abstract ideas (*Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility*, p. 58).

Regarding independent claim 7, for similar reasons, independent claim 7 is also not statutory. Claim 7 also recites a *programmable object model* as well as *...an object-oriented message call; passing an object property to the XML schema library...; and in response to the message call and the object property passed to the XML schema library, receiving access to the functionality identified in the XML schema library associated with the object property passed to the XML schema library*. Claim 7 claims nonfunctional descriptive material for the reasons explained above. Further, while claim 7 represents a collection of abstract ideas, even if the invention were recorded on some

computer readable medium the invention would lack practical application since the result of the invention is *receiving access to the functionality identified in the XML schema library associated with the object property passed to the XML schema library.*

In regard to dependent claims 2, 4, 6, and 8-18, claims 2, 4, 6, and 8-18 are rejected because they add nothing to render the claimed subject matter statutory.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fordin, "Java Architecture for XML Binding: Executive Summary" (hereinafter "JAXB"), Sun Microsystems, published July 2003, p. 1-7, in view of Gertner et al. (hereinafter "Gertner"), U.S. Pub. No. 2003/0135825, published July 2003.**

Independent claim 1 cites: *A programmable object model for accessing the resources of an Extensible Markup Language (XML) schema library, comprising: an application programming interface for allowing a user to programmatically access resources identified in an XML schema library; the application programming interface comprising a message call for requesting association of an XML schema file to an XML markup applied to a document; and the application programming interface operative to*

receive a return value from the XML schema library responsive to association of the XML schema file to the XML markup applied to the document.

JAXB teaches a binding framework which was a runtime application programming interface for accessing resources of an XML schema (p. 4, par. 4). JAXB teaches a message call associating an XML schema file to an XML markup applied to a document (See Figure "Steps in the JAXB Binding Process", p. 5, and p. 5-6 "The JAXB Binding Process"). JAXB teaches validation of XML document content via the JAXB binding framework, compare to *receive a return value from the XML schema library responsive to association of the XML schema file to the XML markup applied to the document.* While JAXB does not explicitly teach an XML schema library, Gertner teaches using a programmable object model, SOX (p. 2, par. 22), to programmatically access an XML schema library (p. 2, par. 24; Fig. 2; p. 3, par. 35). Both JAXB and Gertner are analogous art because both are directed toward the dynamic generation of web services. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Gertner to JAXB, since JAXB provides a framework to assist developers by allowing them to easily take advantage of XML technologies (JAXB, p. 1, par. 5) and Gertner would give JAXB the benefit of a method for generating a web based Graphical User Interface (GUI) thereby providing a tool to assist developers in the creation of XML technologies (Gertner, p. 1, par. 5-7).

Regarding dependent claim 2, JAXB teaches an unmarshalling process which may or may not require validation against the source schema, and can also take the form of streams of data passed between applications (p. 5, "XML Input Documents"; p.

6, par. 2-7). JAXB teaches custom binding declarations which may be overridden (p. 6-7). Compare to claim 2, *a message call for requesting removal of an association of the XML schema file to the XML markup applied to the document.*

Independent claim 3 cites: *A programmable object model for accessing the resources of an Extensible Markup Language (XML) schema library, comprising: an application programming interface for allowing a user to programmatically access resources identified in an XML schema library; the application programming interface comprising a message call for requesting association of an XSLT transformation to an XML markup applied to a document; and the application programming interface operative to receive a return value from the XML schema library responsive to association of the XSLT transformation to the XML markup applied to the document.*

JAXB teaches a binding framework which was a runtime application programming interface for accessing resources of an XML schema (p. 4, par. 4). JAXB teaches a message call associating an XML schema file to an XML markup applied to a document (See Figure "Steps in the JAXB Binding Process", p. 5, and p. 5-6 "The JAXB Binding Process"). Further, JAXB teaches that data can be mapped to an XML document (p. 3) and the document can be accessed or updated without traversing a DOM tree. For this reason, it would have been obvious to one or ordinary skill in the art at the time of the invention that JAXB could be used to associate XSLT transformations to XML documents, since JAXB discloses the association and mapping of XML document contents and XSL was simply a stylesheet language for XML.

JAXB teaches validation of XML document content via the JAXB binding framework. While JAXB does not explicitly teach an XML schema library, Gertner teaches using a programmable object model, SOX (p. 2, par. 22), to programmatically access an XML schema library (p. 2, par. 24; Fig. 2; p. 3, par. 35). Gertner also teaches generating XSLT stylesheet transformations to the generated markup document (p. 3, par. 29). Both JAXB and Gertner are analogous art because both are directed toward the dynamic generation of web services. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Gertner to JAXB, since JAXB provides a framework to assist developers by allowing them to easily take advantage of XML technologies (JAXB, p. 1, par. 5) and Gertner would give JAXB the benefit of a method for generating a web based Graphical User Interface (GUI) thereby providing a tool to assist developers in the creation of XML technologies (Gertner, p. 1, par. 5-7).

Regarding dependent claim 4, JAXB teaches an unmarshalling process which may or may not require validation against the source document, and can also take the form of streams of data passed between applications (p. 5, "XML Input Documents"; p. 6, par. 2-7). JAXB teaches custom binding declarations which may be overridden (p. 6-7).

Regarding independent claim 5, independent claim 5 is directed toward substantially similar subject matter as claimed in independent claim 1, and is rejected along the same rationale, because claim 5 cites *association of an one or more XML-based resources to an XML markup applied to a document,* and claim 1 cites

association of an XML schema file to an XML markup applied to a document. An XML schema file comprises at least one XML-based resource.

Regarding dependent claim 6, dependent claim 6 is directed toward substantially similar subject matter as claimed in dependent claim 2, and is rejected along the same rationale.

Independent claim 7 cites: *A programmable object model for accessing the resources of an Extensible Markup Language (XML) schema library, comprising: calling the XML schema library via an object-oriented message call; passing an object property to the XML schema library, the object property being associated with a software object associated with functionality identified in the XML schema library; and in response to the message call and the object property passed to the XML schema library, receiving access to the functionality identified in the XML schema library associated with the object property passed to the XML schema library.*

JAXB teaches a binding framework which was a runtime application programming interface for accessing resources of an XML schema (p. 4, par. 4). JAXB teaches a message call associating an XML schema file to an XML markup applied to a document (See Figure "Steps in the JAXB Binding Process", p. 5, and p. 5-6 "The JAXB Binding Process"). JAXB discloses calling an XML schema via an object oriented message call and passing object properties to the schema, since JAXB binds XML schemas to Java object oriented representations (P. 1, "What is JAXB?").

While JAXB does not explicitly teach an XML schema library, Gertner teaches using a programmable object model, SOX (p. 2, par. 22), to programmatically access an

XML schema library (p. 2, par. 24; Fig. 2; p. 3, par. 35). Both JAXB and Gertner are analogous art because both are directed toward the dynamic generation of web services. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Gertner to JAXB, since JAXB provides a framework to assist developers by allowing them to easily take advantage of XML technologies (JAXB, p. 1, par. 5) and Gertner would give JAXB the benefit of a method for generating a web based Graphical User Interface (GUI) thereby providing a tool to assist developers in the creation of XML technologies (Gertner, p. 1, par. 5-7).

Regarding dependent claim 8, JAXB teaches passing a method for creating a new XML Namespace and for adding the new XML Namespace to a collection of XML Namespaces, where a path to a schema file associated with the new XML Namespace and a uniform resource identifier for the new XML Namespace are passed to the XML schema library as parameters of the method object (p. 6, "How Does JAXB Represent XML Content?").

Regarding dependent claim 9, JAXB teaches binding customizations of the XML namespace URI (p. 6-7). It would have been obvious to one of ordinary skill in the art at the time of the invention that the binding customization which includes package naming, would result in the registration of namespaces, since a customized URI would require registration of the new namespace.

Regarding dependent claims 10-12, JAXB teaches a name-mapping algorithm to label schema components, and includes a set of type safe enum classes and a dynamic instance factory allocator and instance factory method for each Java element

interface in the package (p. 6, "Binding XML Names to Java Identifiers" and "Java representation of XML Schema"; also see p. 5, "Steps in the JAXB Binding Process").

Regarding dependent claims 13-15, JAXB teaches that the binding framework was typically wrapped in a larger Java application that would provide XML transformation functions (p. 5, par. 1). While JAXB does not explicitly teach passing an object property that points to a default XSLT transformation associated with a specified Namespace, Gertner teaches the use of schema adjuncts which were name/value pairs tied to a specific portion of the schema, and which included XML Namespace objects (p. 3-4, par. 38-39). Gertner teaches associating schema adjunct information with stylesheets which removed the need to bind specific presentation information to each application page (p. 4-5, par. 42-43). Gertner teaches binding method properties to a schema library.

Both JAXB and Gertner are analogous art because both are directed toward the dynamic generation of web services. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Gertner to JAXB, since JAXB provides a framework to assist developers by allowing them to easily take advantage of XML technologies (JAXB, p. 1, par. 5) and Gertner would give JAXB the benefit of a method for generating a web based Graphical User Interface (GUI) thereby providing a tool to assist developers in the creation of XML technologies (Gertner, p. 1, par. 5-7).

Regarding dependent claims 16 and 17, JAXB teaches a name-mapping algorithm to label schema components, and includes a set of type safe enum classes and a dynamic instance factory allocator and instance factory method for each Java

element interface in the package (p. 6, "Binding XML Names to Java Identifiers" and "Java representation of XML Schema"; also see p. 5, "Steps in the JAXB Binding Process"), i.e., a numerical index where the index is passed with the method property. While JAXB mentions accessing XSLT transformations with Java, and teaches programmatic access to individual XML elements, JAXB does not explicitly teach accessing individual XSLT transformations. However, Gertner teaches using schema adjuncts to access stylesheet elements, where elements can be identified by name (p. 4, par. 41-42). Gertner teaches that categories of adjuncts are identified using a namespace prefix (p. 4, par. 39-40). Both JAXB and Gertner are analogous art because both are directed toward the dynamic generation of web services. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Gertner to JAXB, since JAXB provides a framework to assist developers by allowing them to easily take advantage of XML technologies (JAXB, p. 1, par. 5) and Gertner would give JAXB the benefit of a method for generating a web based Graphical User Interface (GUI) thereby providing a tool to assist developers in the creation of XML technologies (Gertner, p. 1, par. 5-7).

Regarding dependent claim 18, JAXB teaches allowing the developer to choose which elements to bind to classes and how to bind each attribute and element declaration to a property in the appropriate content (p. 7, par. 1-4).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amelia Rutledge whose telephone number is 571-272-7508. The examiner can normally be reached on Monday - Friday 9:30 - 6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AR

HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100